Fundusze Europejskie
Wiedza Edukacja Rozwój

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Społeczny

Politechnika Śląska jako Centrum Nowoczesnego Kształcenia
opartego o badania i innowacje

POWR.03.05.00-IP.08-00-PZ1/17

Projekt współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego

Microprocessor and Embedded Systems

Faculty of Automatic Control, Electronics and Computer Science, Informatics, Bachelor Degree

Lecture 15

Motorola 68xx-based microprocessors and microcontrollers

Bartłomiej Zieliński, PhD, DSc

68xx

Program:

- A little history
- 6800 and its derivatives
  - 6809
  - 6502
  - 68HC05, 08, 11
  - ST7LITE
  - Nitron

68xx

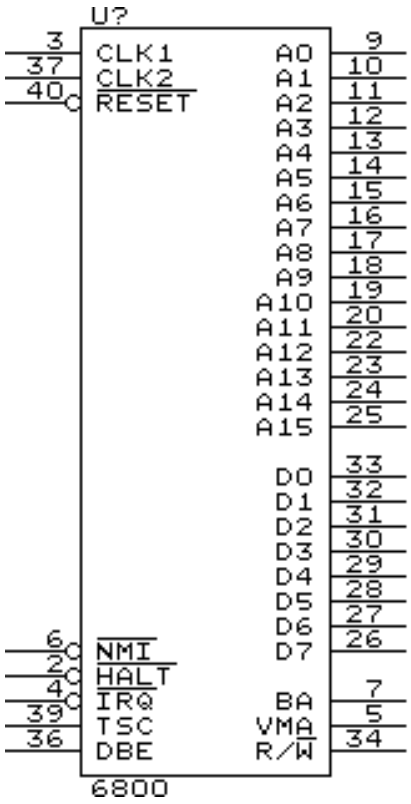
- Motorola 6800
  - Designed in 1974
  - Based on DEC PDP-11 architecture
  - 16-bit address bus
  - 8-bit data bus
  - No separate I/O space
  - 72 commands, 197 opcodes
  - F_{max}=1÷2 MHz
  - Single power voltage (rare in this time)

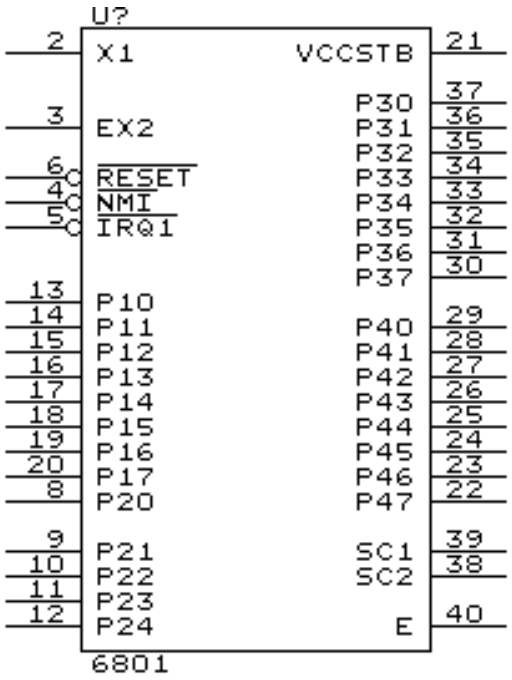
68xx

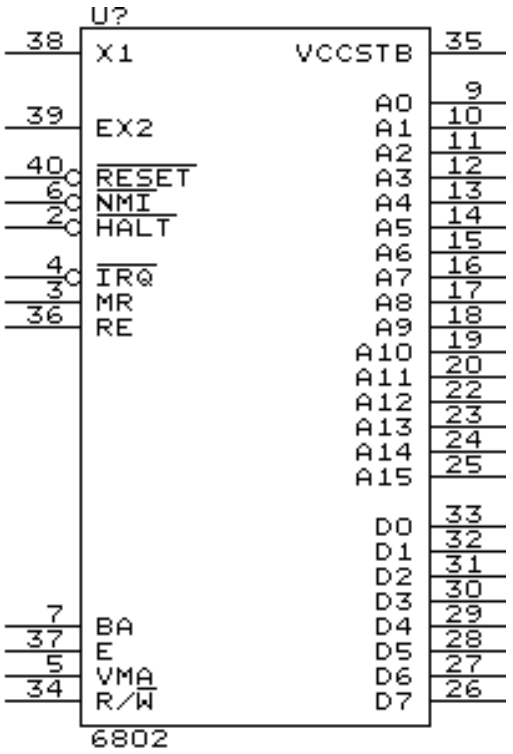
- Motorola 6800 family
  - Peripherials
    - 6810 RAM 128×8
    - 6820 PIA (Peripherial Interface Adapter)
    - 6830 ROM 1024×8
    - 6850 CIA (Communications Interface Adapter)
    - 6860 Modem
    - 6870 Two-Phase Clock
  - µp/µc
    - 6802 – includes RAM and clock
    - 6801, 6805 – RAM, ROM, peripherials

68xx

- Motorola 6800

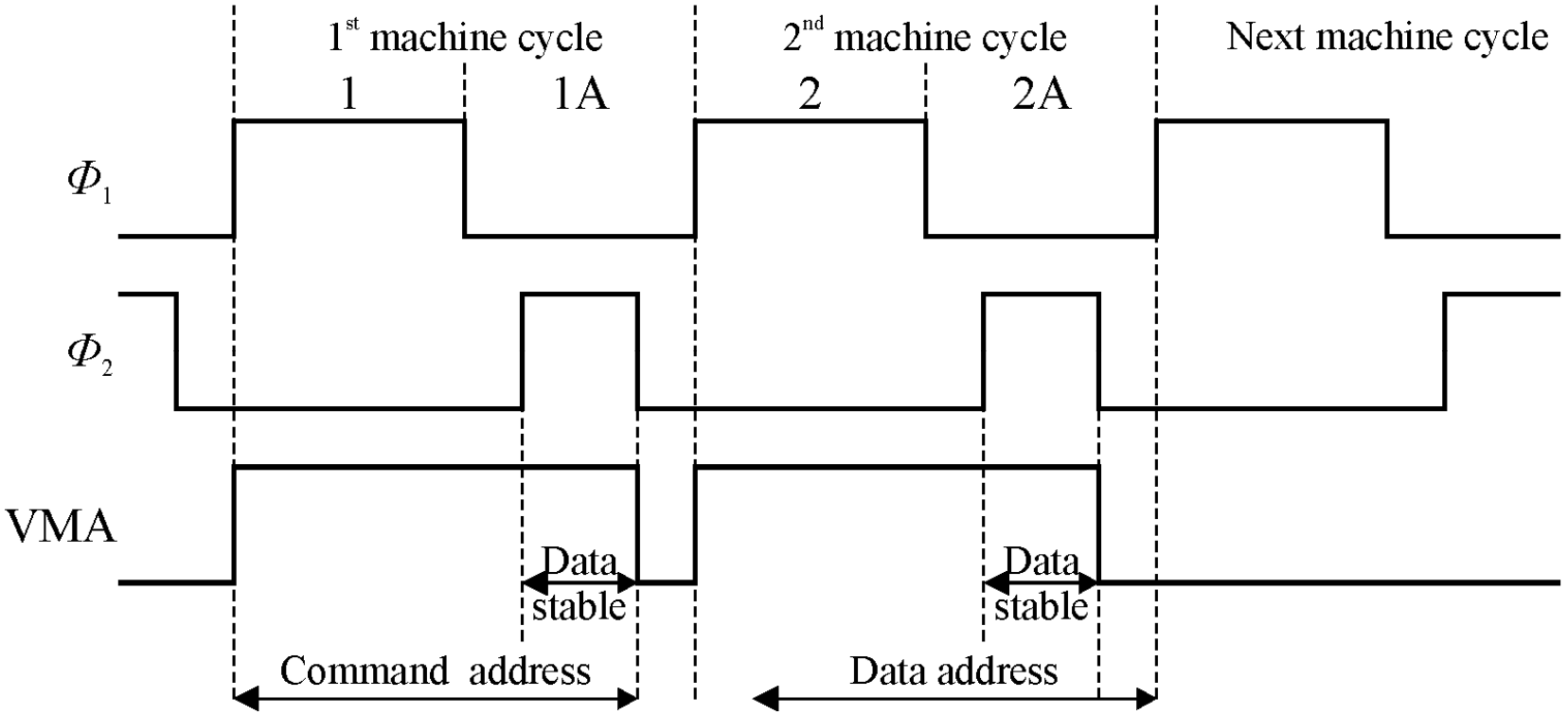






68xx

- Motorola 6800 clock

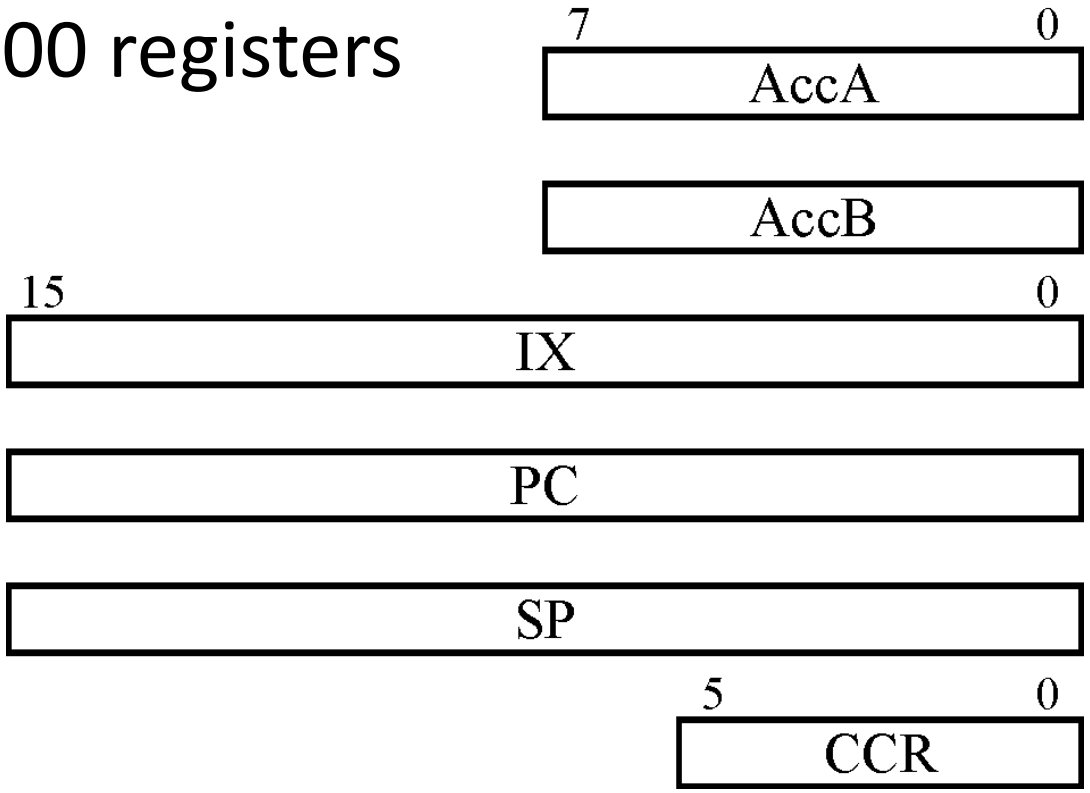


68xx

- Motorola 6800 control signals
  - $R/\overline{W}$
  - VMA – Valid Memory Address (1 a moment after the address is placed on the bus)
  - $\overline{\text{Reset}}$ – (reset vector FFFE, FFFF)
  - $\overline{\text{NMI}}$ – (NMI vector FFFC, FFFD)
  - (SWI) – (SWI vector FFFA, FFFB)
  - $\overline{\text{IRQ}}$ – (IRQ vector FFF8, FFF9 and lower)
  - $\overline{\text{Halt}}$ – finish command & halt; A&D buses Hi-Z
  - BA – Bus Available; µp is halted
  - TSC – Tri-State Control (1→tri-state Address bus)
  - DBE – Data Bus Enable (0→tri-state Data bus)

68xx

- Motorola 6800 registers



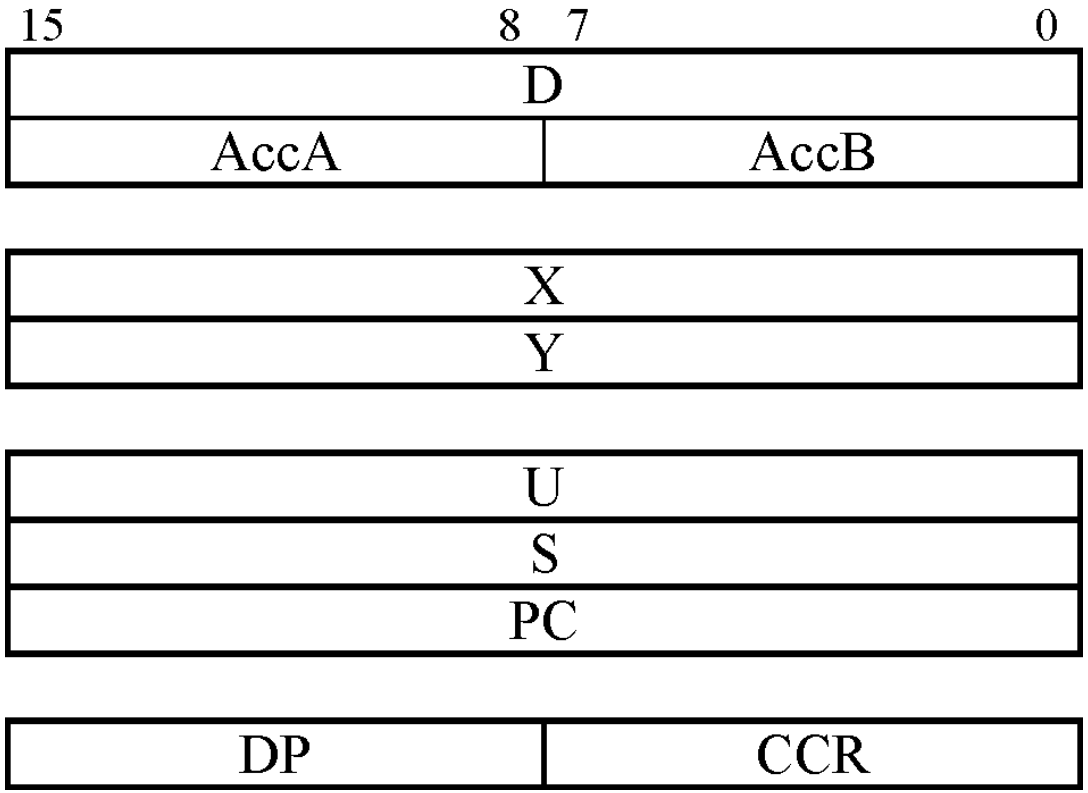
  - CCR (flags)
    - Half carry, Interrupt, Negative,
    - Zero, oVerflow, Carry

68xx

- Motorola 6809
  - Designed in 1978
  - 16-b extensions
    - Mul/div command
  - Two 8-b accu's can be merged in a single 16-b acc
  - Two 16-b index registers
  - User and System stack pointers
  - Command list simplified
    - 59 commands vs. 78 in 6800
    - Source-code compatibility

68xx

- Motorola 6809 registers

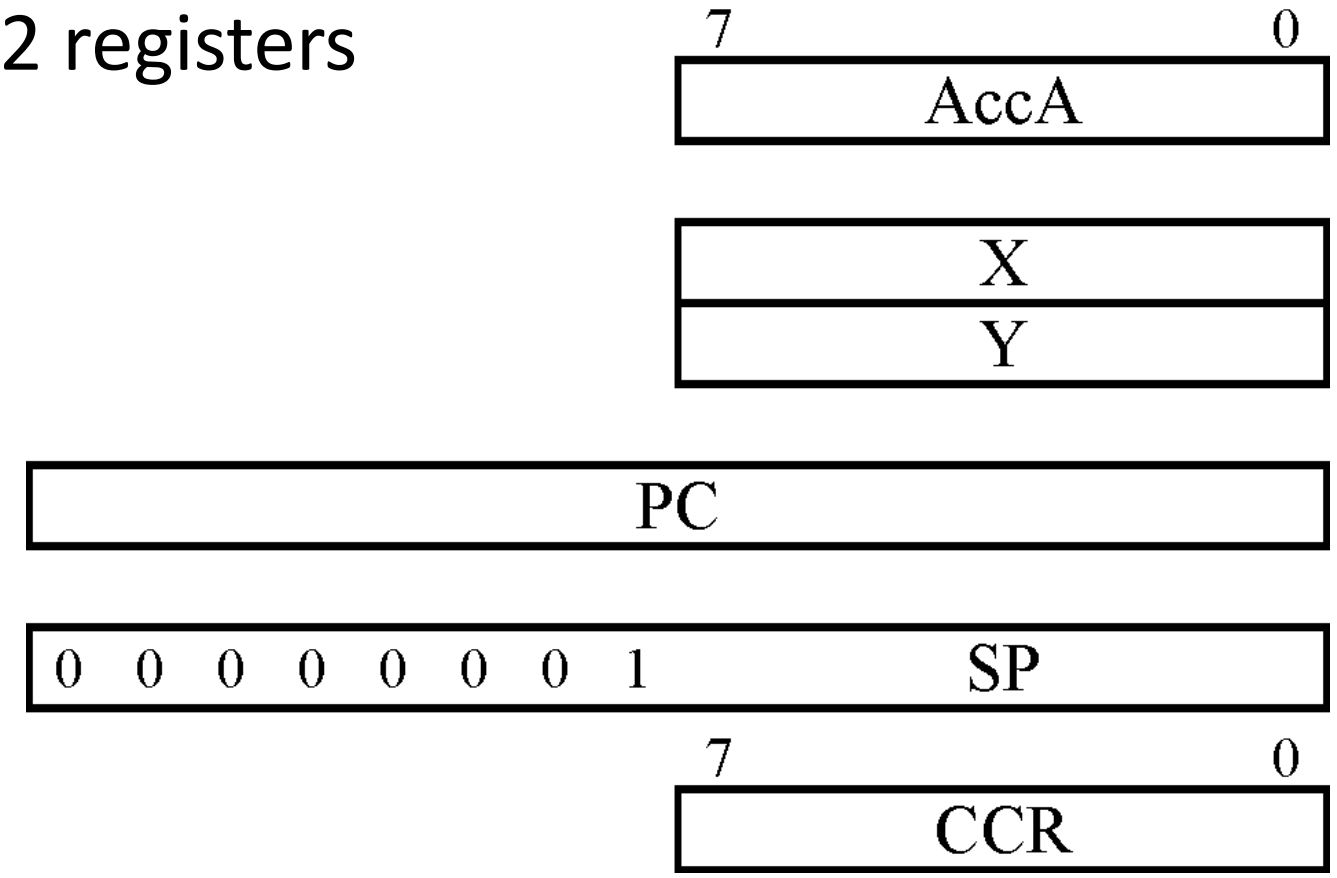


68xx

- MOS Technology („Mostek”) 6502
  - Designed in 1975
    - +/- the cheapest on the market
    - With Z80 began home computer revolution
  - Similar to 6800 design, simplified
    - No tri-state buses
    - Only a single accumulator
    - Stack limited to 256B
    - Index registers limited to 8-b
      - Base address stored in command
    - Command set simplified (56 commands)
    - 6501 was 6800-pin compatible (lawsuit with Motorola)

68xx

- MOS 6502 registers



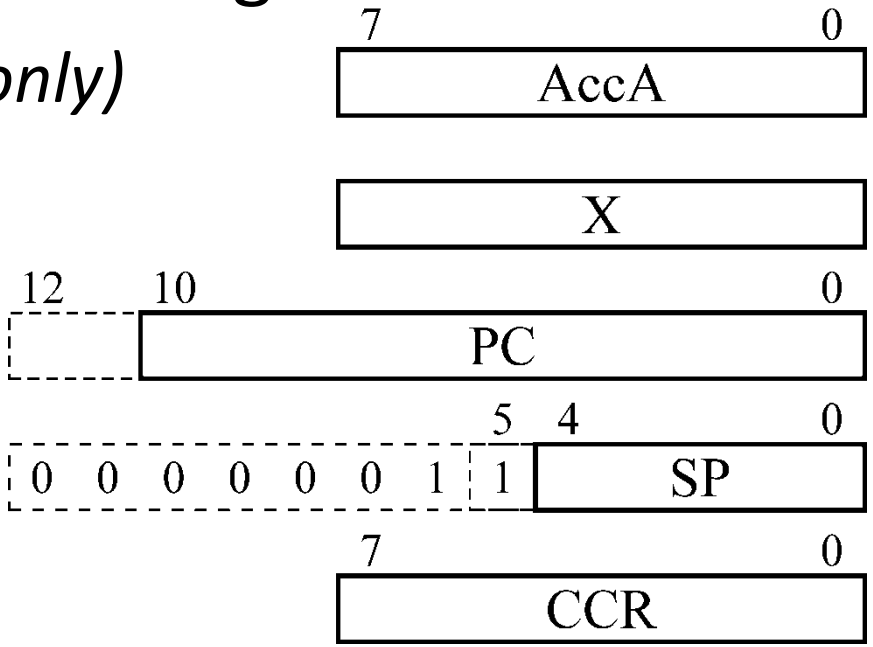
68xx

- Motorola 68HC08 family
  - Full executable-code compatibility with HC05
    - Including only 8-b of X register pushing onto a stack
  - Extended command list
    - 78 new opcodes
    - Fast mul/div, BCD, mem→mem transfer, etc.
    - New addressing modes
  - Extended address bus support
    - Memory page switching
  - DMA support
    - Only the most complex µc

68xx

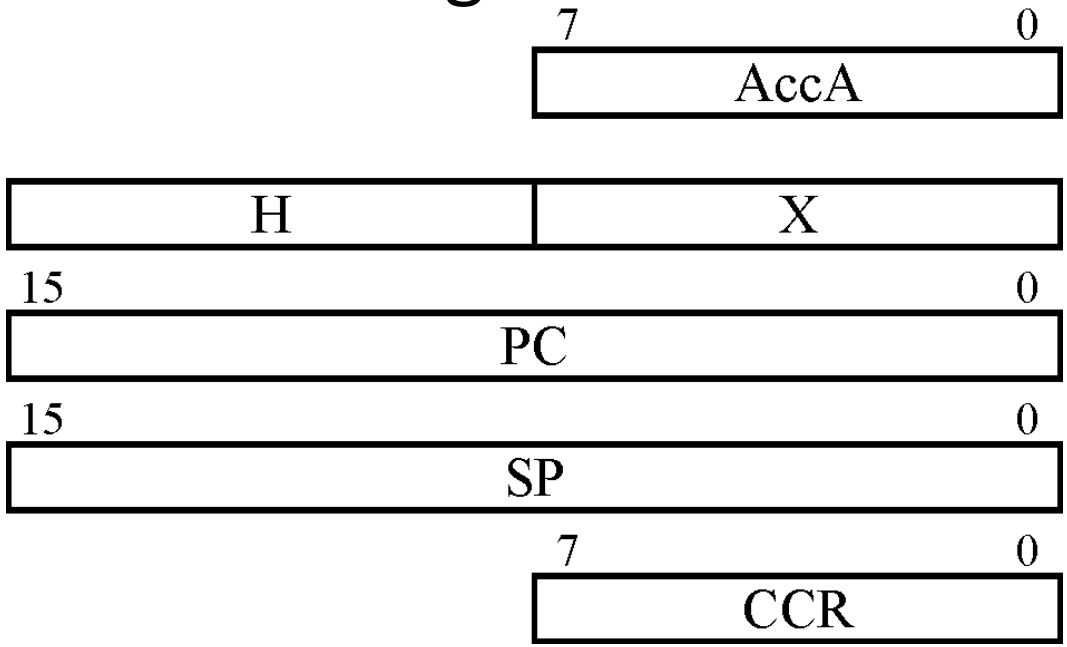
- Motorola 68HC05 registers

  (for reference only)



68xx

- Motorola 68HC08 registers



  - For compatibility with HC05:
    - Reset → H=0
    - Reset → SP=00FF

68xx

- Motorola 68HC08 addressing modes
  - Implicit/inherent
  - Immediate
  - Direct
  - Absolute/extended
  - Relative
  - Indexed without offset
  - Indexed with 8/16-bit offset
  - *Stackpointer Indexed with 8/16-bit offset*
  - *Memory-to-memory (4 modes)*
  - *Indexed with post increment*
  - *Indexed with 8-bit offset and post increment*

68xx

- Motorola 68HC08 addressing modes
  - Implicit
    - No explicit argument
    - Mostly 1-byte opcode
    - Commands executed within 1 clock period
    - Executed on A, X, H:X
    - Very effective, very fast
    - Limited by low register count
    - Few examples:
      - `Inca/Deca, incx/decx`, …
      - `Psha/pula, pshh/pulh`, …
      - `nop`

68xx

- Motorola 68HC08 addressing modes
  - Immediate
    - 8/16-b argument directly after opcode
      - Argument is not and adress
    - To write a constant into a register
      - 16-b regs: H:X, SP, PC only
    - To compare register with a constant
    - Few examples:
      - `Cmp #nn; compare A with #nn`
      - `Ldx #nn; x=#nn`
      - `Ldhx #nnnn; h:x=#nnnn`
      - `Add #nn; a+=#nn`
      - `Cbeqa #nn; jump if a=#nn`

68xx

- Motorola 68HC08 addressing modes
  - Direct
    - 8-bit address directly after opcode
    - Higher half of address = 0
      - „page zero addressing”
    - „page 0” → frequently used variables, etc.
      - Faster program execution
      - Lower memory consumption
    - Few examples
      - `Add nn; a+=(nn)`
      - `And nn; a&=(nn)`
      - `Cbeq nn; jump if a=(nn)`

68xx

- Motorola 68HC08 addressing modes
  - Absolute/extended
    - 16-b arg addr directly after opcode
    - 3 bytes: opcode, MSB, LSB
    - Entire memory addressing space can be reached
      - If „page zero” not large enough
    - Programmer free from addressing mode choice
      - Choice is done by the assembler
    - Few examples:
      - `Add nnnn; a+=(nnnn)`
      - `Sta nnnn; (nnnn)=a`
      - `Lda nnnn; a=(nnnn)`

68xx

- Motorola 68HC08 addressing modes
  - Indexed
    - Without offset (1B: opcode)
      - Effective adress = H:X
      - E.g., `jmp ,x; jump to (h:x)`
    - With 8-b offset (2B: opcode, offset)
      - Effective address = H:X + offset
      - E.g., H:X = table base address, offset = table index
      - E.g., `jmp $ff, x; jump to (h:x)+$ff`
    - With 16-b offset (3B: opcode, MSB, LSB)
      - Effective address = H:X + offset
      - E.g., offset = table base address, H:X = table index
      - E.g., `jmp $10ff, x; jump to (h:x)+$10ff`

68xx

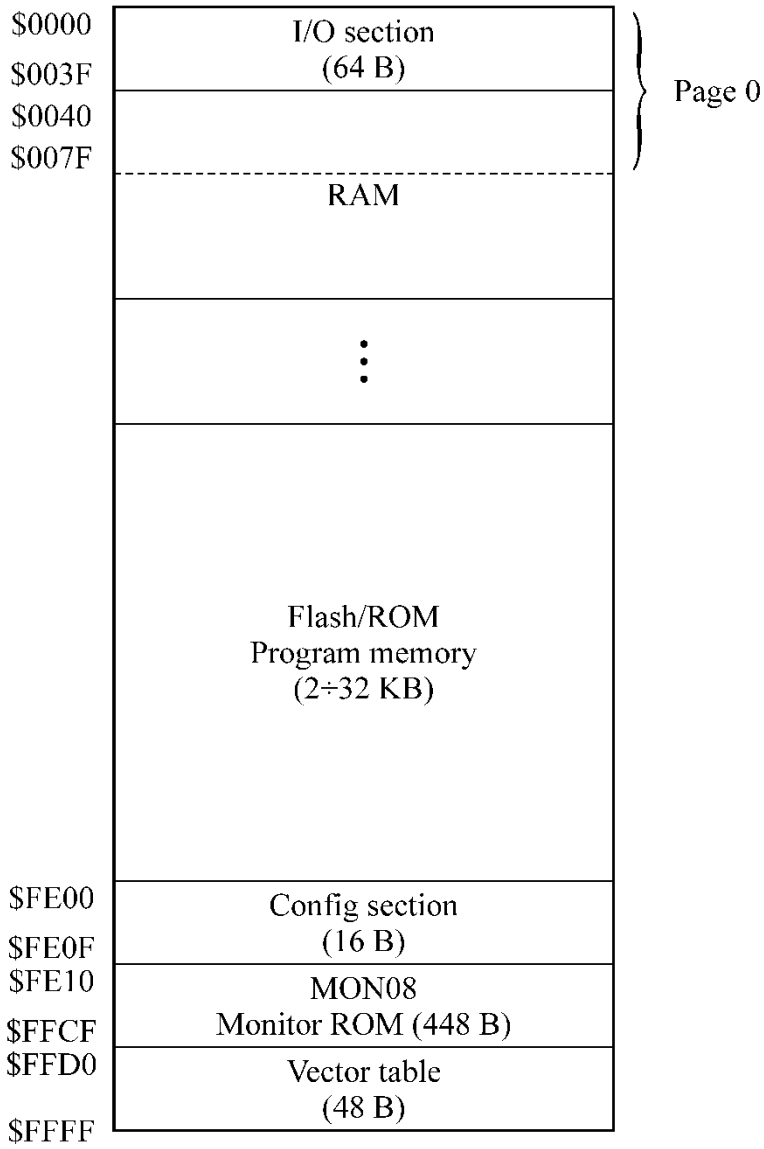
- Motorola 68HC08 addressing modes
  - Indexed
    - With post increment
      - Effective address = H:X; H:X++
      - E.g., `cbeq X+, rel8`
    - With 8-b offset and post increment
      - Effective address = H:X + offset; H:X++
      - E.g., `cbeq ofs8, X+, rel8`
  - Stackpointer Indexed
    - Similar to indexed
    - Interrupts disabled → SP = 2nd index reg
    - All commands must be prefixed (1 byte longer)

68xx

- Motorola 68HC08 addressing modes
  - Memory to memory
    - `Mov` instruction only
    - Copy without using internal registers
    - (new comparing to HC05)
    - Immediate to direct
      - `Mov #nn, ad; (ad)=#nn`
    - Direct to direct
      - `Mov ad1, ad2; (ad1)=(ad2)`
    - Direct to indexed with post increment
      - `Mov ad, X+; (ad)=(H:X++)`
    - Indexed to direct with post increment
      - `Mov X+; ad; (H:X++)=(ad)`

68xx

- General memory map
  - I/O section – peripherials
  - RAM – part in Page 0
    - Typ. 128B÷2KB
  - Flash/ROM
    - End at $FFDF
    - Begin depends on size
  - Config section
    - Specific functions
  - Monitor
    - For debugging
  - Vector table



68xx

- Interrupts

Priority	Source	Flag	Mask	Int reg.	Vector
	Reset	–	–	–	$FFFE-$FFFF
	SWI	–	–	–	$FFFC-$FFFD
	IRQ1	IRQF1	IMASK1	IF1	$FFFA-$FFFB
					$FFF8-$FFF9
	TMR ch. 0	CH0F	CH0IE	IF3	$FFF6-$FFF7
	TMR ch. 1	CH1F	CH1IE	IF4	$FFF4-$FFF5
	TMR ovl	T0F	T0IE	IF5	$FFF2-$FFF3
	Keyboard	KEYF	IMASKK	IF14	$FFF0-$FFF1
	ADC	C0C0	AIEN	IF15	$FFDE-$FFDF

68xx

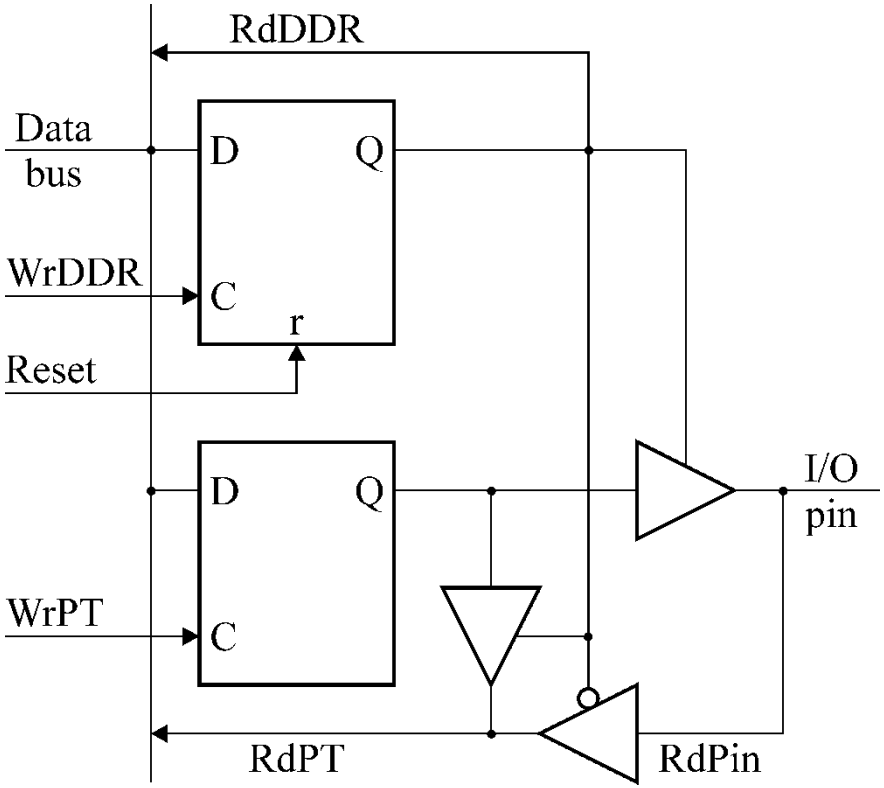
- Interrupts
  - Reset
    - Reasons
      - $\overline{\text{RST}}$ signal – *reset*
      - POR – *power-on reset*
      - LVI – *low voltage inhibited*
      - COP – *computer operating properly*
      - ILOP – *illegal opcode reset*
      - ILAD – *illegal address reset*
    - Action
      - SP=$00FF; H=0; CCR.I=0 (interrupt disable)
      - Remaining registers=random
      - Peripherials initialise
      - PC=M[$FFFE-$FFFF]

68xx

- Interrupts
  - Software interrupt
    - SWI command
    - Non-maskable (does not depend on CCR.I)
    - Possible applications:
      - Testing & debugging – breakpoint
      - Call a procedure by a jump table (like 8086)

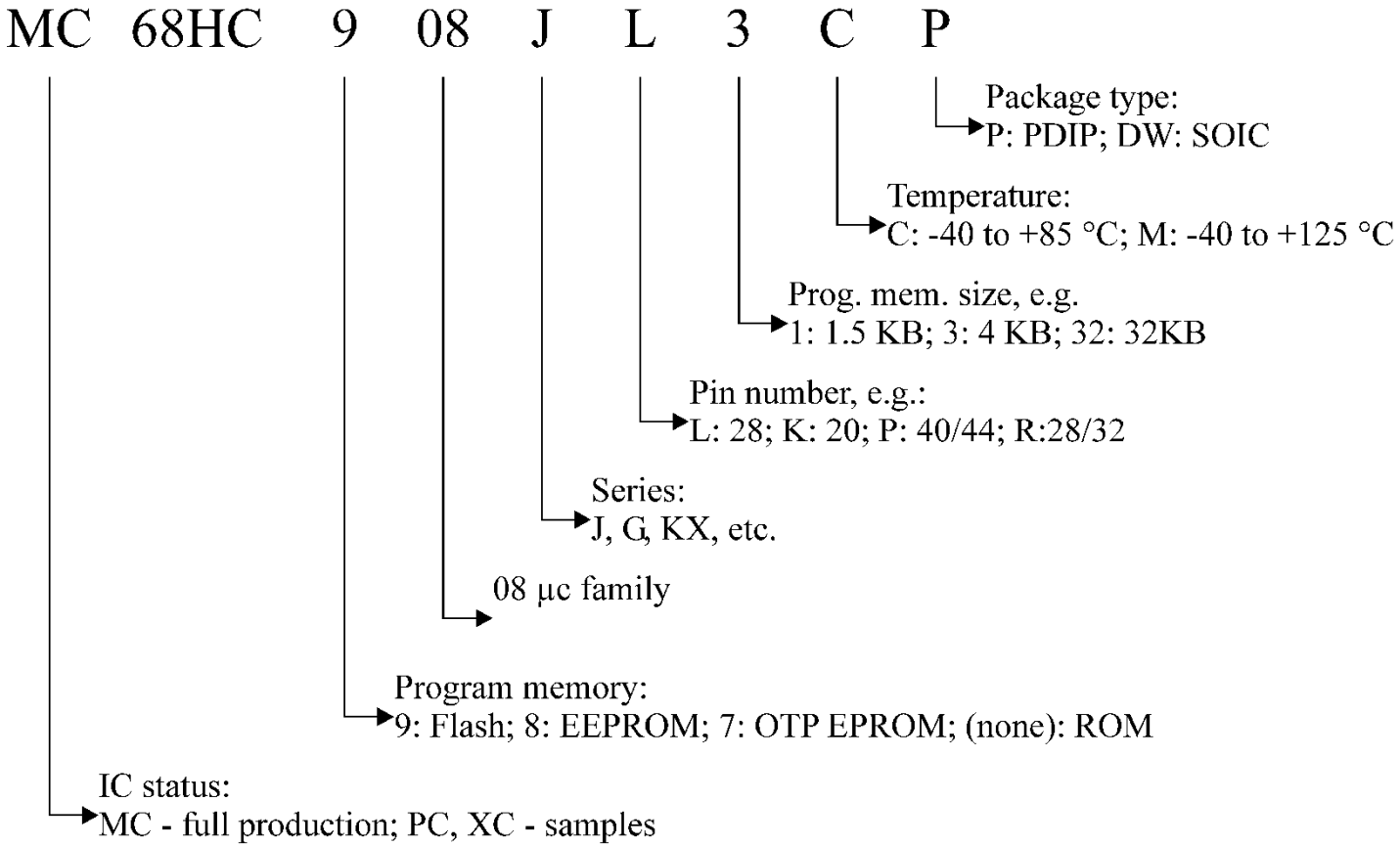
68xx

- I/O ports
  - General structure
    - DDR (*Data Direction Register*)
      - 0 = input, 1 = output
    - PT (*Port Register*)
  - Possible options
    - Pull-up resistors
    - Open-drain output
    - High current out.
    - Peripherial I/O
    - Etc.



68xx

- Circuit names
  - E.g., MC68HC908JL3CP

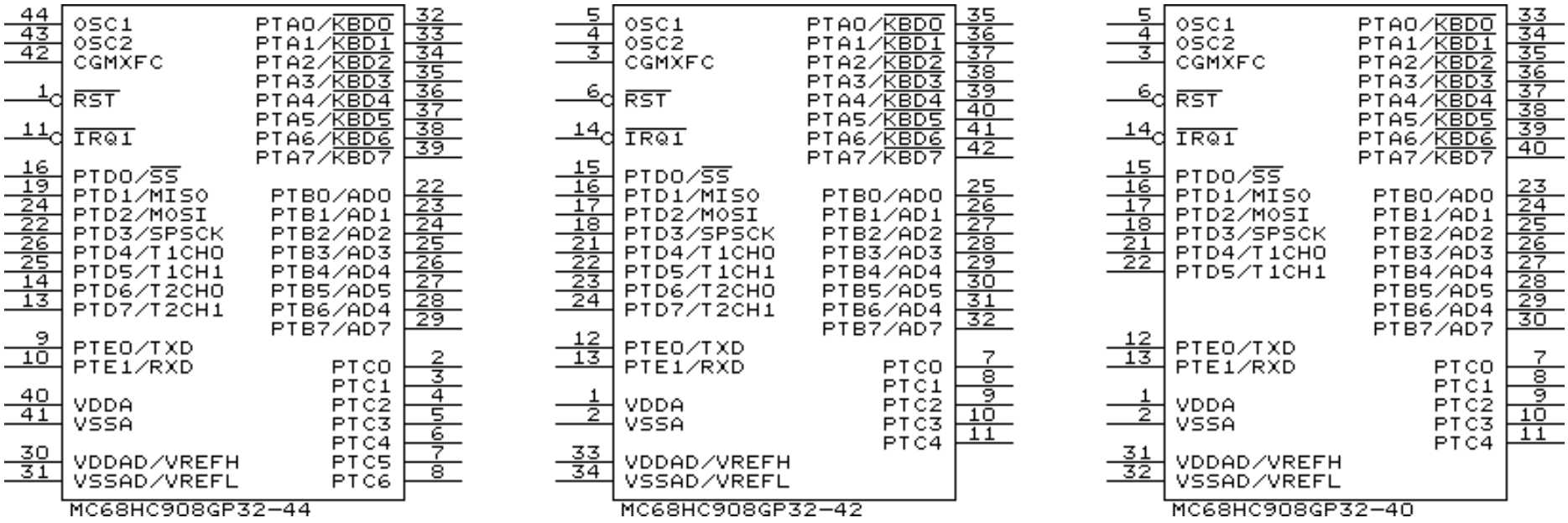


68xx

- µc series
  - J – low cost (20, 28 pins)
  - G – general purpose (28/32, 40/44 pins)
  - KX – „small” applications (16 pins)
  - AZ – with CAN interface (QFP 64/100 pins)
  - RK – „*remote keyless entry*” (20 pins)
  - JB – with USB (DIP 20, SOIC 28, QFP 44 pins)
  - MR – for electric motor control (QFP 64 pins)
  - etc.

68xx

- GP series pinout examples



  - If pins not available in some package
    - PTC5, PTC6 – connected to gnd
    - PTD6, PTD7 – unconnected

68xx

- J, KX, QT, QY series pinout examples